

WHAT IS CLAIMED IS:

1. A method for verifying a software upgrade for a communication device, comprising:

5 determining a plurality of transient variables maintained by a communication device executing first software;

10 comparing a first variable set maintained by the communication device executing first software to a second variable set maintained by the communication device executing second software to identify changed variables; and

15 indicating a failure of the verification if at least one of the changed variables does not match one of the transient variables.

2. The method of Claim 1, wherein the step of determining a plurality of transient variables comprises:

20 determining a first copy of a variable set maintained by the communication device executing first software;

determining a second copy of the variable set maintained by the communication device executing first software; and

25 comparing the first copy to the second copy to identify transient variables.

3. The method of Claim 1, further comprising:

repeating the steps of determining, comparing, and indicating for a predetermined number of counts; and

30 indicating a pass of the verification if a failure has not been indicated after the predetermined number of counts.

4. The method of Claim 1, wherein the step of comparing comprises:

loading the communication device with first software;

5 determining a first variable set;

loading the communication device with second software;

determining a second variable set; and

10 comparing the first variable set to the second variable set to identify changed variables.

5. The method of Claim 1, wherein the steps of determining, comparing, and indicating are performed by a software management system remotely coupled to the communication device using a packet network.

6. The method of Claim 1, wherein the first and second variable sets comprise at least a portion of a managed information base (MIB) maintained by the communication device.

7. The method of Claim 1, wherein generating a first variable set comprises:

25 issuing a query to the communication device for a specified variable;

receiving a value for the specified variable from the communication device; and

repeating the steps of querying and receiving for each variable maintained in the first variable set.

8. A software upgrade verification system, comprising:

an interface operable to couple to a communication device;

- 5 a processor coupled to the interface and operable to determine a plurality of transient variables maintained by the communication device executing first software, the processor further operable to compare a first variable set maintained by the communication device executing
- 10 first software to a second variable set maintained by the communication device executing second software to identify change variables, the processor further operable to indicate a failure of a software upgrade verification if at least one of the changed variables does not match
- 15 one of the transient variables.

9. The system of Claim 8, wherein, to determine the transient variables, the processor is further operable to:

- 20 determine a first copy of the variable set maintained by the communication device operating first software;

determine a second copy of a variable set maintained by the communication device operating the first software;

25 and

compare the first copy to the second copy to identify the transient variables.

10. The system of Claim 8, further comprising a memory operable to store first software and second software, wherein the processor is operable to:

- retrieve first software from the memory;
- 5 communicate first software to the communication device using the interface;
- receive a first variable set maintained by the communication device executing first software using the interface;
- 10 retrieve second software from the memory;
- communicate second software to the communication device using the interface;
- receive a second variable set maintained by the communication device executing second software using the
- 15 interface; and
- compare the first variable set to the second variable set to identify the changed variables.

11. The system of Claim 8, wherein the interface is
- 20 operable to couple to the communication device using a packet network.

12. The system of Claim 8, wherein the first and second variable sets comprise at least a portion of a
- 25 managed information base maintained by the communication device.

13. The system of Claim 8, wherein the processor is further operable to:

issue a query to the communication device for a specified variable using the interface;

5 receive a value of the specified variable from the interface;

store the specified value in the memory; and

repeat the query, receive, and store operations for each of a plurality of variables in the first variable
10 set.

14. Logic encoded in media for verifying a software upgrade for a communication device, the logic operable to perform the following steps:

5 determining a plurality of transient variables maintained by a communication device executing first software;

10 comparing a first variable set maintained by the communication device executing first software to a second variable set maintained by the communication device executing second software to identify changed variables; and

15 indicating a failure of the verification if at least one of the changed variables does not match one of the transient variables.

15 15. The logic of Claim 14, wherein the step of determining a plurality of transient variables comprises:

20 determining a first copy of a variable set maintained by the communication device executing first software;

determining a second copy of the variable set maintained by the communication device executing first software; and

25 comparing the first copy to the second copy to identify transient variables.

16. The logic of Claim 14, further comprising:
repeating the steps of determining, comparing, and
indicating for a predetermined number of counts; and
indicating a pass of the verification if a failure
5 has not been indicated after the predetermined number of
counts.

17. The logic of Claim 14, wherein the step of
comparing comprises:
10 loading the communication device with first
software;
determining a first variable set;
loading the communication device with second
software;
15 determining a second variable set; and
comparing the first variable set to the second
variable set to identify changed variables.

18. The logic of Claim 14, wherein the steps of
20 determining, comparing, and indicating are performed by a
software management system remotely coupled to the
communication device using a packet network.

19. The logic of Claim 14, wherein the first and
25 second variable sets comprise at least a portion of a
managed information base (MIB) maintained by the
communication device.

20. The logic of Claim 14, wherein generating a first variable set comprises:

issuing a query to the communication device for a specified variable;

5 receiving a value for the specified variable from the communication device; and

repeating the steps of querying and receiving for each variable maintained in the first variable set.

21. An apparatus for verifying a software upgrade for a communication device, comprising:

means for determining a plurality of transient variables maintained by a communication device executing first software;

means for comparing a first variable set maintained by the communication device executing first software to a second variable set maintained by the communication device executing second software to identify changed variables; and

means for indicating a failure of the verification if at least one of the changed variables does not match one of the transient variables.

22. The apparatus of Claim 21, and further comprising:

means for determining a first copy of a variable set maintained by the communication device executing first software;

means for determining a second copy of the variable set maintained by the communication device executing first software; and

means for comparing the first copy to the second copy to identify transient variables.